ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

ALL CONDITION WEB COMPOSITION FRAMEWORK (ACWCF)

G.V.SAM KUMAR[1], S.S.RIAZ AHAMED [2]

Assistant Professor, Department of Information Technology, Sathyabama University, Chennai, India[1]

Principal, Sathak Institute of Technology, Ramanathapuram, Tamilnadu, India [2].

ABSTRACT: Current Web services are created and updated on the run. Web service mining is a relatively new area of research. Web services present an entirely new area in research. It is process aimed at discovering interesting and useful compositions in existing Web services. The Web service composition approaches are very useful where primary goals are unavailable or unknown. It differs from the conventional top down approaches driven by specific a criterion. The service does not assume any prior knowledge and instead of searching specific compositions, it relies on component services to explore data and the result may vary from a simple composition to a complex one. Business establishments realized the potential of the Web and took advantage of its popularity by sharing their data and applications on the Web. The web from its starting position as s a repository of information with text and images, evolved into a host for providing multimedia content and service providing application like map-finding, weather-reporting, e-commerce etc. Real-time applications involved hardware devices like temperature sensors and traffic monitoring cameras. Business and government integrated existing Web applications to provide new value-added services. Customized interfaces required applications to access this data and the lack of semantics in data made integration of the applications a overwhelming challenge. This Paper presents a novel approach in Web service mining by introducing a Web service.

I. INTRODUCTION

Current Web services are created and updated on the run. Web service mining is a relatively new area of research. This growing popularity of the Web and Web services presents an entirely new area in research. It is process aimed at discovering interesting and useful compositions in existing Web services. The Web service composition approaches are very useful where primary goals are unavailable or unknown. It differs from the conventional top down approaches driven by specific a criterion. The service does not assume any prior knowledge and instead of searching specific compositions, it relies on component services to explore data and the result may vary from a simple composition to a complex one. Though, the Web was invented in the 1990's by Tim Berners-Lee for information sharing among scientists [1], only after 1993 the Web was available for public information sharing due to the Web browser Mosaic. Mosaic quickly captured the imagination of the public on the Web through its graphical user interface. Governmental Organizations, Business establishments realized the potential of the Web and took advantage of its popularity by sharing their data and applications on the Web. The web from its starting position as s a repository of information with text and images, evolved into a host for providing multimedia content and service providing application like map-finding, weather-reporting, e-commerce etc. Real-time applications involved hardware devices like temperature sensors and traffic monitoring cameras. Business and government integrated existing Web applications to provide new value-added services. Customized interfaces required applications to access this data and the lack of semantics in data made integration of the applications a overwhelming challenge. Attempts to solve these problems resulted in underlying and enabling technologies. Many of the researches are driven by cross-enterprise workflow and AI planning. This Paper presents a novel approach in Web service mining by introducing a Web service. A set of algorithms and a conceptual nut novel framework is also introduced for service compositions.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

The effectiveness of the framework and challenges addressed by the framework are also discussed. The Paper also discusses transactional approaches [2].

II. RELATED WORK

Data mining is extracting interesting information from data in databases. It is the sifting through databases to find patterns of data using computer programs. Data mining receives attention due to the voluminous data made available. Enterprises maintain terabytes of data due the availability of cheap storage media. [3]. Data mining techniques classify data, detect anomalies and predict data. The Data Mining techniques search for consistency in patterns and relationships between variables [4]. The results are then applied on new data sets for validating them. Data mining techniques have been used to forecast weather [5], predict stocks [6], sports analysis [7], medical diagnosis [8], and fraud detection [9]. Data mining is similar to Web service mining in extracting previously unknown and useful information. Web services encapsulate behaviors using a set of dynamic operations and traditional data mining techniques cannot be used to determine the composability of two services. Service composition can be from two fundamentally different strategies. The first is the top down strategy requiring a specific goal based search criteria to start the composition process like traditional Web service composition approaches. Since user provides the goal the type of composition is anticipated by the user. The evaluation of composition interestingness is not a major concern area in such approaches. The bottom up strategy is driven by the need to find interesting and potentially useful compositions of existing Web services and without using specific search criteria (Web service mining). Web service mining techniques need to address the interestingness of service compositions. Semantics-based composition approaches use ontologies to an advantage while describing Web services. The use of ontologies allows unambiguous interpretation of a Web service semantics by a computer. Based on the DARPA Agent Markup Language Ontology Inference Layer (DAML+OIL), Web Ontology Language for Services (OWL-S) [10]. Service ontology aims at enabling Web services composition planning , reasoning and automatic use of services by software agents. Figure 4 shows the elements of a web composition model. Researchers have addressed problems of interleaving web service discovery and compositions by considering simple workflows of web services with one input and one output parameter [11]. The web service composition is restricted to a sequence of limited web services corresponding to a linear workflow of web services. The solution retrieves a sequence of links between web services, to generate a composite service plan from existing services, a composition path was proposed in [12]. The path consisted of a sequence of operators computing data, and connectors with provision for data transport between operators. The search for operators to construct a sequence is based on the shortest path algorithm. Only two kinds of services operator and connector, with an input and output parameter are considered, contrary to the model proposed with more than one input and output parameter [13,14]. A composition of services as a directed graph with nodes linked by matching compatibility between input and output parameters is considered in [15],.The shortest sequence of web services are derived from the graph. The sequence corresponds to an ordered set of web services matching all expected output parameters for the given inputs by a user. Semantic web service compositions are performed by pre-computing the causal link matrix in [16]. The composition strategy is based on AI planning and performs a regression-based approach and returns a set of correct, complete and consistent plans. The services are actions semantically linked by causal links. The composition strategy is based on AI planning and performs a regression-based approach and returns a set of correct, complete and consistent plans. The services are actions semantically linked by causal links. However, these two approaches compute the best composition based on the semantic similarity of output and input parameters of a web services and not considering any non-functional properties [17].A modelling tool called interface automata was introduced to represent web services and perform compositions where Atomic services are stored as a graph and each node represents input and output parameters while edges represent web services. Each service has a description of inputs, outputs, and dependencies in other web services. The service descriptions and a graph used to discover composition results that satisfy a service request. A that composer that supports the end user to select web services for each activity in the composition, create flow specifications to link them is introduced in [18]. After selecting a web service, the web services producing an output are fed as the input of a selected service based on profile descriptions. The user manually selects the service for a particular activity and the system generates a composite process in DAML-S. The composition calls each service separately thus passing the results between services based on flow specifications.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

In Web Service Compositions, Several standardization and prototype efforts were undertaken [19]. Composition related approaches can be grouped into two different categories business process-oriented and semantics-based [20]. The Petri-net approach [21] graphically represents represent operations as a process and a connected graph where nodes (places) are used to represent states and other nodes (transitions). One token in every place connected to an operation enables the operation. The operation may remove one token from every input and deposit the token as output .A service can be in one of the following states: not instantiated, ready to execute, executing, suspended or completed, At any given time. After each service is defined, a variety of compositions can be defined by including sequence, alternatives, iterations etc. A process can also be analyzed in many ways using Petri-nets due to the abundance of analysis techniques [22, 23,24]. Petri-nets can be used to determine the presence of live locks. Algebraic Process Composition models processes are based on calculus [25], in which the basic entity can be one of the following,: an empty process, a choice between I/O operations, a parallel composition, a recursive definition or invocation. I/O operations can receive or send. IBM's Web Services Flow Language (WSFL) [26] is an XML based language for describing Web service compositions. WSFL is based on Petri-nets and provides two models, the flow model and global model. The flow model aims at specifying the logic of a business process. It uses a directed graph to model the sequence of the functionality provided by a composed service to control flow and data between component services. Each node in the graph is an activity and represents a step in the business goal the composition tries to achieve. The control links type uses the links to connect activities and prescribe the activities order. The second type called data links, represent the flow of data between activities. The global model aims at defining the mutual exploitation of Web services in decentralized or distributed business processes. Since no specification of an execution sequence is provided, it relies on the use of plug links to represent interactions. WSFL also aims at supporting recursive compositions of services. In WSFL, every Web service composition (flow, global) can transform itself into a new Web Service and be a component for new compositions. Table 1 lists a comparison between different web compositin technologies.

Table 1. Comparison of BPEL4WS, BPML, WSCI, WS-CDL and DAML-S

	BPEL4WS	BPML	WS-CDL	WSCI	DAML-S
Modeling the collaboration	Strong support	Indirect support	Strong support	Strong support	Strong support
Modelling the execution control	Strong support	Strong support	No support	No support	Strong support
Reprensentation of the Role	Weak support	No support	Strong support	Strong support	No support
Transaction and Compensation	Indirect support	Strong support	Indirect support	Strong support	Indirect support
Exception handling	Strong support	Strong support	Support	Strong support	Strong support
Semantic support	No support	No support	No support	No support	Strong support
Business agreement support	No support	No support	No support	No support	No support
Software vendor support	Many	Few	No	Few	Few

III. WEB SERVICE MINING TECHNIQUES

Web services are being deployed at an accelerating rate. New services are created by extending or combing existing services. Process mining is considered an extension of service mining, due to the inter dependence

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

between business process and web services. Specific queries discovering patterns for a competitive advantage in a business environment may be unavailable. Such mining is a bottom up search process proactively targeting potentially interesting and useful Web Services from existing ones. Systems demand a sophisticated design to answer to user needs and requirements with reliable executions. It is important to track Web services utilization. Business opportunities can be discovered by analyzing different logs and tracking Web Service Interactions with autonomous parties. This chapter compares different web mining approaches for discovery, conformance and extension of examined properties and behaviors and algorithms to extract process trace data from the process logs to develop a novel [27]. Models aim to improve processes used in business process mining.

- COMPOSITE PATTERNS EXTRACTION FROM EXECUTION LOGS: The service composition re-usage of patterns provides an efficient way to improve the quality of new applications. The service pattern is identified by locating associated services commonly used by different applications and understanding control flow in the set of associated services. The application infrastructure facilitates the monitoring of services-oriented applications from execution logs. Composite service patterns constitute requirements of multiple organizations with a specific control flow and effects best practices. Re-using services pattern provides good quality composite web services. Service pattern composition can be built in two ways.
  - Top-Down: The business processes from different organization are reviewed to identify patterns
  - Bottom-UP: Execution logs of applications are analyzed to mine business patterns like frequently executed service patterns. This pattern mining of execution logs can be broken in three sub tasks.
  - Pre-processing: A service-oriented application is executed as multiple instances where each Instance is identified by a unique identifier. Different types of events of the instances are logged. Events like resource adaptor event, business rule event and service invocation event.etc. The entry and exit points of a process are logged in the application logs along with its instance identifier and time stamp. Logs are processed to filter events.
  - Identifying frequently associated services: Services which occur frequently are considered for service pattern. The number of services to be analyzed is pre-defined.
  - Recovering the control flow: The control flow of a service in the service pattern is reusable. The executing instance of a service is considered and execution flow is extracted and similar execution flow is extracted for all services. The Common execution flows in these services is considered for a control flow.
- WEB SERVICE INTERACTION MINING: Business Process Execution Language (BPEL) standardizes web service compositions into business processes. BPEL defines and monitors workflows. Web Service Interaction Mining (WSIM) as an extension to BPEL[28]. WSIM proposes three levels of abstraction on performance. The event log is mined to get information about the web service behaviors, reducing the amount of data. Information on interactions between the web services determines critical dependencies. The transactions are categorized into four types: One way, notification, response for request and solicit response. One-way and notification operations are messages between the sender and receiver. The sender knows the receiver, but the receiver does not have knowledge about the sender. Request-response and solicit response are messages exchanged by the sender and receiver. The initiator sends a message and the called service replies to the message.
- LOG BASED WEB SERVICE MINING: A web service application accessible to customers is identified by a URI with its interfaces and bindings described in an XML document. The service is discovered by other web services and interactions between web services happen with SOAP, UDDI and WSDL. The execution log is mined and analyzed for Web services behavior, which contain many levels due to complexity and the richness of the composition in Web services model. In Level-1 the set of operations offered by a web services are gathered. Level-2 defines additional to set of operations and Level-3 has

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

the set of interactions exchanged within a given choreography. Level-4 defines the composite Executable process implementing a composite Web service. The mining combines data mining techniques on web services log to discover knowledge from the web service model [29]. The first step is gathering data for analysis and gets useful information about web services behavior. The logging is done at two levels Trivial and Advanced..

- WEB SERVICES USAGE MINING: Learning the usage sequence of users from web services can give important information. The information can be used to find better web services by applying web mining techniques to analyze patterns in behavior of web services. Frequent occurrence of a web services could be mined using AprioriAll algorithm. Optimizations for faster timing in the execution of the mining algorithm can be done. The correlations between operations and web services can be discovered using a proper log format. The log must contain start time, connection time, disconnection time, session-id, user-id, service with operations. The log sequential pattern is fed as the input with a defined threshold (occurrence count a web service). The k-item set of operation sets are generated and by reducing size of the candidate set the speed is improved. For Example filtering unrelated services log from the data set.
- PROCESS MINING: Process mining is the discovery or verification on the conformance of processes based on event logs, when Web services are distributed amongst different parties. Process mining helps monitoring the exact execution of processes and determines bottlenecks, unused paths and verifies deviations. The sequence of events is recorded for every process instance called a trace. The event log contains a set of process instances with various properties of processes associated. When the events are not correlated to the process instances, the execution/ monitoring of processes and Key Performance Indicators measurements go wrong. Analysis of Process Mining is useful when the web services are distributed over autonomous parties and they show an emerging behavior. The mining can be of three types, Discovery when no prior model exists and a new model needs to be constructed using event logs. Conformance is done when a prior model exists. The third type of Extension happens when a prior model exists, but is extended with new perspective for enriching the model. IBM Web Sphere Business Monitor satisfies the three types of process mining and thus Process mining techniques can be used to discovery, confirm and extend the existing web services.
-

IV. COMPOSITIONS IN WEB MINING

Web today is a collection of Web Services. A number of companies have projects involving Web services and growing further. Web Services are Open standards with widespread support for universal access in a neutral Platform. Web service to the learner is promising technology allowing sharing individual and autonomous software. Surveys show application Integrations or Web Service Compositions are gaining primary importance. Application integration costs consume Twenty Five percent of the total IT budget in many companies. New systems can also be designed by composing from existing software components, accessible as web services. This reuse of components lowers maintenance costs. Web services composition is a challenging problem as the number of service provider's increase. Further, many of the applications may overlap in functionality and content and dis-jointed. With web services, the functionality and content can be shared and remotely accessed. Any modular approach in information systems engineering breaks down a system into parts where the parts are individually designed, constructed, tested and often multiple service providers need to co-ordinate to create a new service. This service-oriented paradigm enables creation of services that are well-described, implemented independently and interoperable. The role of Developers also differs in creating new systems by combining available web services provided by service providers, resulting in web compositions. Even if a single service cannot serve user needs, a combination of services can solve the problem. The term Semantic Web is applied to an extension of current Web in which information is given a well-defined meaning. Semantic Web services are designed to support automatic discovery, invocation or interoperations. Service interfaces, discovery and service invocation are performed using XML based standards called WSDL, UDDI and SOAP [30]. Web Service Compositions are presently based on Workflows, XML or Ontology based. The Web service models consist of three parts, the service provider, the registry and the consumer [31]. The use of standard protocols, SOAP, WSDL and UDDI helps integrate a composite service irrespective of

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

platforms or execution speeds. Web service composition has also received increasing attention from the research community [32].

A.. Requirements for Web Service Compositions

The main aim of a service composition is to support the process of connecting different services from vendors to execute a specific task. Decisions about services selection, deployment are taken during the composition process. Each stage of a composition has different requirements on methods. In the orientation phase the cooperation is described in general terms and an overview of available of services may not be available. In the negotiation services needed are decided in co-operation with the parties offering the services and their specifications. The compositions of the negotiation phase can be realized subsequently in the usage phase. The usage phase starts the composition and ends when one of the actors/services ends the co-operation. Reaching an agreement regarding a composition is the crucial step in the process. The parties decide on functions to be executed in case of a disaster. Figure 5 depicts the stages of a composition process and Table 4 lists the differences between composition standards.

Table 2. Comparison between Web Service Compositions

	BENCHMARKS				
Approaches	CONNECTIVITY	EXCEPTION HANDLING	SCALABILITY	CORRECTNESS	QOS
E - Flow	Low	Low	Low	Low	Low
PPM	Low	Low	Low	Low	Low
BPEL4WS	Good	Good	Average	Low	Average
OWL - S	Good	Average	Good	Low	Good
WSMO	Good	Good	Good	Low	Good

V. FRAMEWORKS FOR WEB SERVICE COMPOSITIONS

Compositions may have to incorporate information from various sources and domains to achieve a task in a Service-Oriented Architecture (SOA). One Single web service is the basic unit of operation in a SOA. One single web service may not be able to satisfy the functional requirements. Web services can also be logically connected to satisfy complex functional requirements. Traditional composition approaches involve human involvement, making it time-consuming and error prone. Automatic compositions compose services into a coherent task which may not be available readily. Different situations require different compositions. It is difficult to find a single web service that would produce the desired output from inputs with user constraints. To create composite services the specification is decomposed into subtasks that eventually correspond to web services, connected with process logic. The Figure 8 is an example of a workflow derived from the high-level specification. The task is subdivided into sub tasks by decomposing the higher-level specification and binding single web services to tasks and executed based on the logic of the workflow. Traditionally, web service composition has been performed manually, making it a difficult and error prone task. This chapter examines various approaches and proposes a general-purpose framework for automatic service composition.

A.. Manual Web Service Composition Process

In a manual composition a user creates an abstract specification of a high-level task with task goals and conditions. The user needs to possess enough domain knowledge to create an abstract composite workflow similar to Figure 8 and by decomposing the specification. When the user lacks domain knowledge a proper work flow is not created. The user utilizes web service standard languages like WS-BPEL [33] or OWL-S [34].to creates the workflow with additional constraints. In the next step, the user binds web services to the workflow. This method is widely used to search and discover services published in a web service repository like UDDI [35]. There is no

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

guarantee that services will match with task descriptions. After binding an executable process is created and has to be monitored to handle faults when they arise. This direct manipulation provides the user with great autonomy and control over the execution. The user has to be knowledgeable in the application domain, web service standards, and security requirements. It is time-consuming and a more error-prone process with no guarantees making it imperative to automate compositions.

B. Automatic Web Service Composition

Automatic web composition can be defined as a five phase process namely specification, planning, validation, discovery and execution. In the Specification phase the User specifies goals, requirements and constraints. The planning phase provides an automatic way to compose an abstract workflow. Validation phase has techniques to ensure realization of composite process. Discovery phase, Discovers services satisfying task specifications and the request is executed finally in the Execution phase.

C. Dynamic Web Services Composition

One great benefit of SOA is, it enables dynamic service binding allowing services to discover, select and invoke services at runtime in Web services technologies [36]. Web services are programmatically accessible over the Internet and interoperate independently and offer flexibility and scalability needed by compositions to profit from the SOA benefits. Automated service discovery, selection and composition are expected to enrich the experience of service end-users through value-added services allowing automated processes to interact with minimal interventions [37]. Dynamic composition of web services is an approach for service oriented applications. It exploits semantic matchmaking between the service parameters outputs and inputs to enable interconnections and interactions. Functional and non-functional properties of services are considered, for the computation of suitable service compositions of a service request. For example a service developer has to develop a new service receiving text for translation and send the translated text to a given place. The developer creates the service by connecting available single web services manually when service is unavailable. In a dynamic web service composition a Composition Factory component creates service compositions based on a service request. The Composition Factory queries the service repository to retrieve an unordered set of services.

VI. ALL CONDITION WEB COMPOSITION FRAMEWORK (ACWCF)

ACWCF is a novel and new direction to composition frameworks. The sections that follow discuss the framework with respect to five stages and can be implemented by organizations based on the guidelines. The user selects a service and the constraints and Qos parameters are trapped along with the request. The required service is searched in the web directory of services and selected. In case the service is not listed in the framework the, an error message is sent to the user and the error is trapped in the error log. One the service is listed and selected, a scope definition is constructed based on the constraints. The scope definition also determines the search space and the real return of information occurs and the results are returned to the user. Whenever an error occurs like in the execution of service or fetching of results or in the network connection, the errors are logged in thye error log giving a scope for improvement of the composition in future versions. The error logging is a simple and useful utility not found in many compositions. The ACWCF can be used by organizations as a starting point for web services compositions and future developments. Figure 12 depicts the framework proposed.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

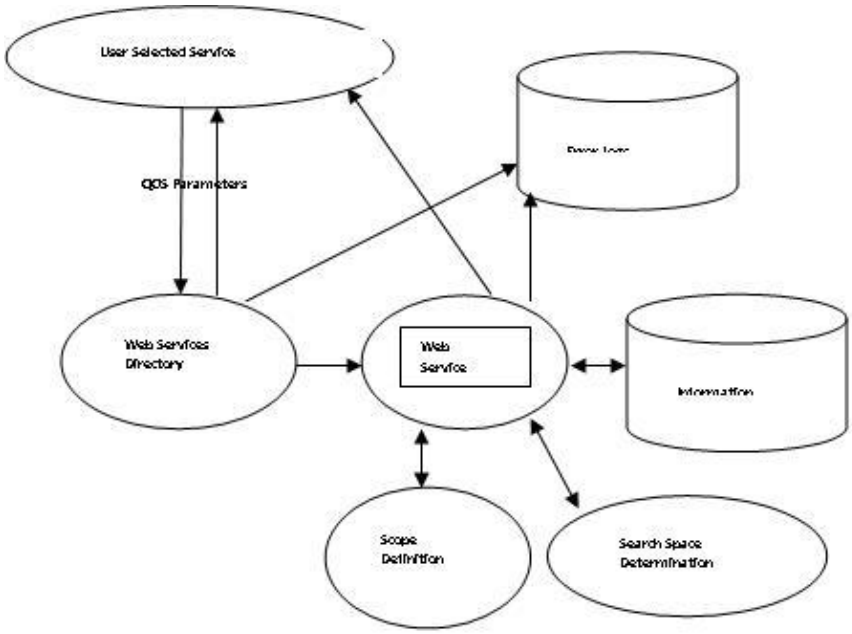

Figure 1. ACCSF

A.. Scope Definition

The mining process starts with the a subjective evaluation of the user request where a the web service takes into consideration the Qos measures specified by the request. The Qos measures are defined as a list of functional activity in the scope specifications of the service. For Example a user request for a cheap rented staying place will have an upper limit of 500 rupees which is taken into consideration in this phase. It would call the service servicing hotel details with a defined upper limit of 500 rupees. For the service engineer developing the service, it becomes mandatory to write functions on the Qos parameter like Cost. Similarly each domain will have separate functional parameters. The subjective evaluation determines a set of ontologies to use for the mining process. The mining context MC can be formally defined as.

$$MC = \{f(Q) \mid d \in D\} \quad (1)$$

where mining context MC , D is a set of Web service domains, Q is a set of Qos attributes of mining interest and f(Q) is the subject evaluated in the domain D. Assume the symbol → denotes refers to and the set of all ontologies are be referred to in MC,can be denoted as O(MC) and calculated using:

$$O(MC) = \{O \mid \exists f \in C \wedge f \rightarrow Ot\} \quad (2)$$

B. Search Space Determination

In any mining operation the scope determines the coverage of the search space. The end product of a search space determination phase is a function library consisting of Web service functions from web service registry WRg that are involved in mining context MC. The function library Li can be defines as .

$$Li = \{s \mid s \in WRg\} \quad (3)$$

Where s is the set of operations of the web service. Algorithm 1 lists the algorithm for obtaining the function library.

C. Algorithm 1 Service Selection

The best web service is executed from the composite web service after selection, optimization and ranking Selection of relevant web services is first performed by matching the web services with the inputs in the request. The inputs are unique for operations and so the required services are selected from the composite.The service selection can be defined as

$$SC = \{f(c) \in C\} = Oc \quad (4)$$

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

where SC is the Services context , C is a set of Web services in the directory, f(c) is the chosen service from a the select list of all services in the Directory

```
Input: Service Selection SC = {f(c) ∈ C}
Output: Selected Service  f(c)
        for all C ∈ SC do
                if  srv = f(Q) do
                        add srv to OC;
                end if
        end for
       if Oc = 0 then
    add to error log
   return failure
end if
return Oc;
```

D. Algorithm 2 Function Selection

```
Input: Mining context MC = {f(Q) | d ∈ D}, Web service registry WRg.
Output: Function library Li.
Variable: Context operations O(MC)
        for all f(Q) ∈ MC do
                for all op ∈ f(Q) do
                        add op to O(MC);
                end for
        end for
        for all s ∈ WRg do
                if s= O(MC) then
add s to Li;
end if
end for
if count of operations cnt = 0 then
    add to error log
   return failure
ebd if
return Li;
```

E. Filtering

Algorithms used in the filtering can be at the operation and parameter level. The operation level interfaces serve as the medium for Web service operations as illustrated in Algorithm 3. Algorithm 4 illustrates a parameter level filtering. The filtering algorithms can be applied to a mining context included in the scope mining.

F. Algorithm 3 operation level filtering

```
Input: Context operation interfaces Opint (MC), function library f(Q).
Output: Leads of composed Web services L.
Variables: Leads from operation interfaces of Opint
for all Opint ∈ Opint (MC) do
List  Service(Opint );
end for
for all s ∈ f(Q) do
        for all op ∈ s.operations do
                if ∃Opint ∈ Opint (MC): op implements Opint then
Lps ← Agent(Opint ).publish(op);
L.add(Lps);
end if
```

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

end for

G. Parameter Level Filtering

Input: context ontologies O(MC), function library f(Q).
Variables: ontology type type(p) p is a parameter .
for all s ∈ f(Q). do
for all op ∈ s.operations do
for all pout ∈ op.messageout do
k ← type(pout);
if k ∈ O(MC)) then
Service(k).select(pout);
end if
end for
end for

H. Relationships between Services

Two services of a web service composition have a relationship if they can be plugged together to perform a value added service or if one of the service can be substituted by the other. Relationships between services can be of the following types.
Service 1 → Service 2 // calls the next service(Prerequisite)
Service 1 // Service 2 // parallel execution
Service 1 ←→ Service 2 // one service substitutes another
Service 1 Ɔ Service 2 // one service includes another
In a Prerequisite Relationship Service 1 has to finish before service 2 starts. In Parallel Relationship Service 1 and service 2 execute in parallel, but the results of each service are combined before further execution. In a Substitute Relationship Service 1 and service 2 can be substituted with each other functionally. In an Include Relationship Service 1 provides services that include the services offered by service 2. The Semantic relationship between two services can be identified by checking the semantic conditions between them. The conditions can be of four types.
Exact match (→)
Plug-in match (PI→)
Plus match (+→)
Complementary match (CP→)
In an Exact Match two Conditions C1 and C2 exactly match denoted by C1 → C2. A Plus-in Match is denoted by C1 PI→ C2 where condition C1 is stricter than condition C2. A plus match also be C1 +→ C2 where condition C1 only partially satisfies condition C2. A Complimentary Match,between two Conditions is denoted as C1 CP→ C2, where condition C1 compliments condition C2. The general approach to create and use web services and compositions is to use JAXRPC and Java RMI services. The next chapter explains in detail the implementations in JAXRPC and RMI services.

VII. DISCUSSIONS AND EVALUATIONS

Several parameters need to be considered while evaluating a Web Composition like connectivity, exception handling, scalability, correctness and Qos attributes of the composition. .
Connectivity: Reliability in connectivity is the basic requirement of a Composition and in its absence even the best composition cannot function.
Exception handling: must deal with exceptional cases of failure in transactions and roll back the transaction to its previous state.
Scalability: Scalability is the ability of the Web service to process multiple requests within a specified time interval. It is measured by the number of requests resolved within a time span and scalability increases with its handling capacity.
Correctness: is a term attached to the behavior of a composition when the service compositions become concurrent or large or complex in nature
QoS: is the requirements parameters from users used to produce the end results in service.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

Though all compositions offer connectivity without which it would be impossible to integrate a set of services, the ability of a service to handle non-functional properties (Qos) is desirable. For example a customer should not wait at the ATM for three whole minutes to get a response. When Compositions neglect non-functional properties like response times, security, costs, reliability and scalability, it is incomplete. In case of correctness of a composition, the service and composition specifications are used to assess and verify its behavior under different circumstances. Automated compositions treated as a part of semantic web vision are intelligent processes with specifications on services and requirements. This design leads to a composition generated without human involvement. All well-defined services should have an unambiguous selection in requirements which is the clear undisputed goal for performance. Performance evaluation of Web services can help implementers understand the behavior of the services in a composed process[38]. The performance of a single Web service can potentially affect the overall performance in a composition. It is customary to evaluate the performance of the services within a process before making it line for usage. A common approach to obtain performance results of a given Web service is by performance testing from a user's perspective the total execution time of a process is a measure of its efficiency. Optimization is a process of selecting services from existing ones in the composite service based on parameters. Ranking is selecting the best web service from web services with highest Performance Index. Services are ranked based on a weighted mean value of the QoS parameters.

A. Time Analysis

The execution time is dependent on three factors Service Time (S), Message Delay Time (M) and Waiting Time (W) [93]. Service Time is the time taken by a Web service to perform a task. Message Delay Time is the time taken to send/receive messages by invocation, determined by the size of the message transmitted/received and the network load Waiting Time is the Web service invocation time delay caused by the load on the system where the Web service is deployed. The Total Invocation Time (T) for a Web service σ is given by the following formula.

$T(\sigma) = M(\sigma) + W(\sigma) + S(\sigma)$

Performed tests on Web service invocations used a ping function for each Web service and by sending and receiving XML messages. Service Time tests were enacted by the environment on load and waiting. Waiting Time tests estimated waiting time by running the test in an environment where the Web service was pre-loaded with requests as displayed in Figure 2. The tested results of BarnesGetPrice service is used for evaluation [39].

10: Num of Iterations

: BarnesGetPrice	:CheckCredit	SendcreditLowInfo
:1.913394	:1.342986	:0.759598
:1.714697	:0.682717	:0.486978
:1.562682	:1.42475	:0.692787
:1.286834	:0.648781	:0.473643
:1.828123	:0.816607	:0.525161
:1.765729	:0.876123	:0.486001
:1.502508	:0.93461	:0.584121
:1.617242	:0.644826	:0.464647
:2.435804	:0.82555	:0.691154
:2.126191	:0.855064	:0.472463
:1.84127	:0.918563	:0.57291
Total :18.330585	:8.622691	:5.399835
Average :1.8330585	:0.8622691	:0.5399835
Minimum :1.502508	:0.648781	:0.464647
Maximum :2.4358	:1.42475	:0.692787
STD :0.276012	:0.2134661	:0.0824082

Figure 2. Test Results for Total Invocation Time

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

Tests on Web services loaded to determine the three time measures Service Time, Message Delay Time, Waiting Time for each service invocation provides analysis on the performance of individual Web services in a composed process. Figure 3 shows the distribution of the overall time for one test where the SendLowCreditInfo Web service had a high Waiting Time indicating that the Web service or the system hosting could not handle the load.

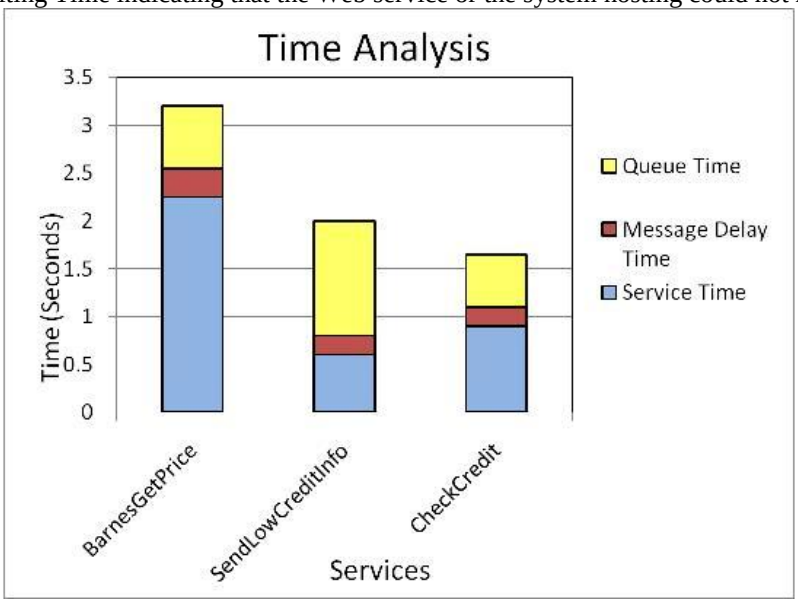

Fig 3. Performance time Analysis

B. Costs Associated Directly with Communications Protocols

When a Web Service application client connects to a service, submits a request and terminates it is unlikely to load server's CPU, or the network. Such services perform well compared to distributed object systems. Table 7 shows relative performances for four implementations an application with a request for a single data structure retrieved according to an argument key AND Figure 4 shows the throughput of different technologies graded against time factor.

Table 3. Costs of single request using various technologies

Technology	Total Latency	Total Packets	Total data transferred in bytes
WS	0.11s	16	3338
CORBA	0.48s	8	1111
CORBA & name server	0.86s	24	3340
Java RMI	0.32s	48	7670

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

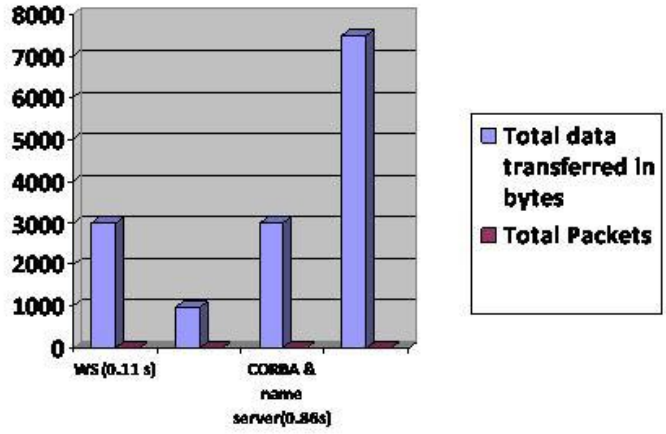

Fig 4. Throughput of Technologies

Remote-connections incur TCP/IP connection costs. Any Web Service using a static stub previously generated from WSDL, incurs the cost of only one connection. The request is sent in two parts (HTTP , SOAP). A response is multipart (HTTP response header, XML response document). Despite the inefficiency of HTTP, the JAXRPC solution performs best. The Java-RMI solution has a complex mechanism; by establishment of a connection to the rmiregistry, then submission for lookup, establishing a connection to an HTTP server, posting a request for a class file and finally exchange with the actual server. RMI protocol entails additional data traffic even in a ping operation. Intranet applications involve a client submitting many requests to a server with additional set-up overheads of the object technologies. Earlier studies studied applications involving multiple requests and comparing average times excluding establishment costs [40, 41]. Table 8 shows results of traffic analysis for technologies. The data shows total byte transfers and total number of packets and averages of repeated tests based on an "iterator". Larger XML documents lead to poorer performance in JAXRPC solutions. JAXRPC solutions require three to five times data packets and from three to ten times in number of bytes. CORBA implementations perform best with the large data transfers. [42, 43]. CORBA has an advantage over Java-RMI primarily because of its use of an 8-bit character encoding for data transfers as opposed to Java-RMI's 16-bit coding. Another factor is Java-RMI's inclusion of meta-data classes in response. Java-RMI solutions perform relatively poorly in terms of total number of packets in the large data. Table 8 shows a traffic analysis.

Table 4. CPU times for client and server with varying technologies and applications

Application	"calculator"		"data"		"large data"	
Technology	Client CPU	Server CPU	Client CPU	Server CPU	Client CPU	Server CPU
WS	15.0s	6S	22.8S	8.4S	1087S	551S (436S)
Java-RMI	2.3S	0.8S	4S	1.1S	148S	212S (97S)
CORBA	3.2S	1.9S	3.6S	2.1S	54.2S	250S (136S)

C. Web Service Results

Figures 5,6 and 7depict the performances of different technologies in web service composition in terms of throughput network conditions and response times.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

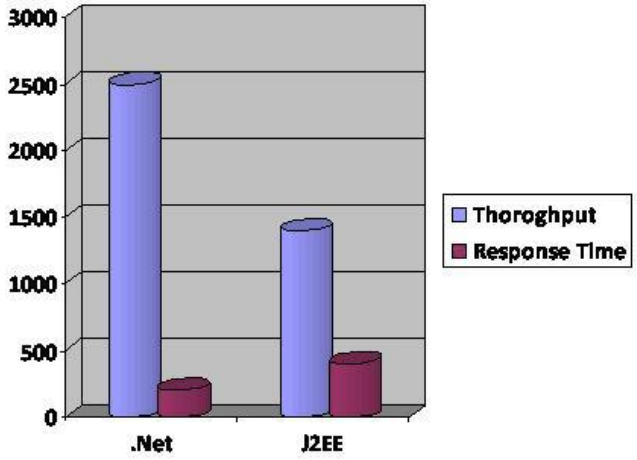

Fig 5. Performance Factors

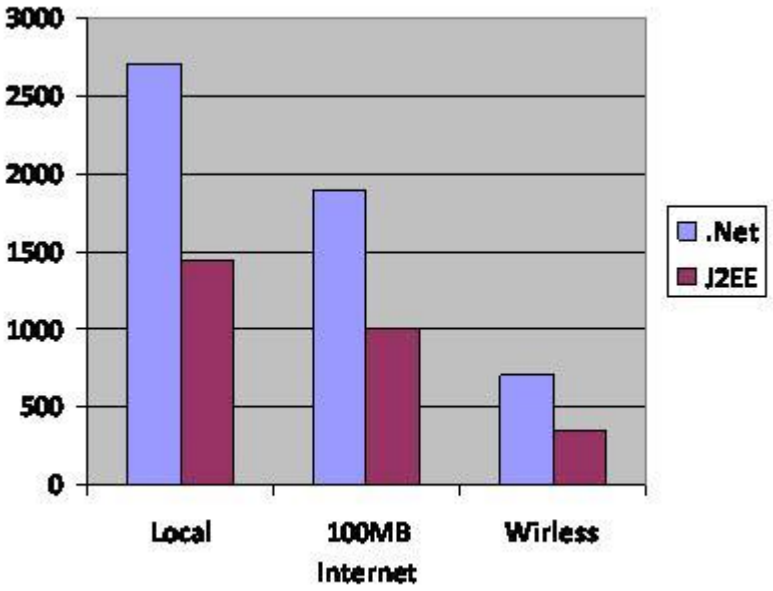

Fig 6. Effect of Network Conditions on Throughput

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

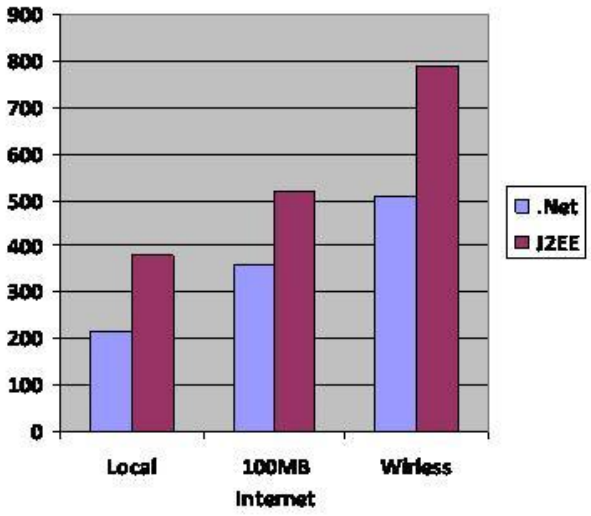

Fig 7. Effect of Network Conditions on Response time

VIII. CONCLUSION

Web services mining has advantages in many fields and more so in the field of business intelligence. Process mining can be used for automatic execution of web services and interoperations between web services. Efficient mining algorithms and techniques are required to analyze and infer from the data collected from process logs. This Paper discusses mining algorithms, describes parameters for efficient processes and a framework for web mining. There are many methods described for web mining but it is the perspective of the user implementing web service mining to use the appropriate algorithm and composition depending on needs of the application implementing the service or composition. Data Filtering is from logs required to remove unwanted data to correlate these and enable data mining tasks like searching of services or failures of processes, thus enhancing the performance, accuracy and reliability of a business process. Manual web service composition can be error-prone and a tedious process. Automatic web service composition frees the configurer from the burdens of manual composition. Real-world applications need to find an acceptable middle stage between manual and automatic composition that relieves the user from the complexity but gives the user enough involvement and confidence for a predicted end result. Though there are many approaches to web service compositions, there is no one perfect solution for all approach. Guidelines on composition processes are also varied. The approaches suggested enable the proactive discovery of interesting pathways for web services. The suggested Web service mining framework allows the mining of interesting composite services to be carried out with the presence of execution logs. The framework can be a base for future works and development. This study aims to give an overview of recent progresses in automatic Web services composition and a five phase model for Web service compositions. The steps are defined as specification, planning, validation, discovery and execution phases. The phases are enabled by workflow research or AI planning. The workflow is applicable to situations for pre-defined models and AI planning methods are applicable for new compositions with user constraints and Qos parameters. Further work will include a detailed analysis of the field in addition to practical testing of and experiments with the methods.

REFERENCES

[1] Weaving the Web - The Original Design and Ultimate Destiny of the World wide Web by Its Inventor. Tim Berners-Lee with Mark Fischetti. HarperSanFrancisco, 1999.
[2] A transactionoriented framework for composing transactional web services. Sami Bhiri, Claude Godart, and Olivier Perrin IEEE SCC, pages 654–663. IEEE Computer Society, 2004.
[3] Tower of power. Information Week, Rick Whiting, February 2002
[4] Stat Soft Inc. Data mining techniques. http://www.uprforum.com/index.html, 2003
[5] Data Mining. Ian H. Witten and Eibe Frank.Morgan Kaufmann Publishers, San Francisco, California, 2000
[6] Building Data Mining Applications for CRM Alex Berson, Kurt Thearling, and Stephen J. Smith.. McGraw-Hill, December 1999.

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

[7] Data Mining: Concepts and Techniques. Jiawei Han and Micheline Kamber. Morgan Kaufmann Publishers, 2000. and electric load fluctuation [Adaptive Computing Laboratories of Iowa. Electric load forecacsting methods. `http://www.aclillc.com
[8] Stat Soft Inc. Data mining techniques. http://www.uprforum.com/index.html, 2003
[9] Data Mining: Concepts and Techniques. Jiawei Han and Micheline Kamber. Morgan Kaufmann Publishers, 2000
[10] W3C. http://www.daml.org/services/owl-s/. 2004
[11] Interleaving discovery and composition for simple workfows. Lassila, O., Dixit, S. In: First International Semantic Web Services Symposium. (2004)
[12] Fault-tolerant, scalable, wide-area internet service composition. Technical Report CSD-01-1129, Mao, Z.M., Katz, R.H., Brewer, E.A .University of California at Berkeley (2001)
[13] Interleaving discovery and composition for simple workfows. Lassila, O., Dixit, S.In: First International Semantic Web Services Symposium. (2004)
[14] Fault-tolerant, scalable, wide-area internet service composition. Technical Report CSD-01-1129, Mao, Z.M., Katz, R.H., Brewer, E.A, University of California at Berkeley (2001)
[15] Automatic composition of semantic web services. Zhang, R., Arpinar, I.B., Aleman-Meza, B., In: 1st International Conference on Web Services. (2003) 38–41
[16] A formal model for semantic web service composition. L´ecu´e, F., L´eger, A., In: ISWC 2006. LNCS, vol. 4273 (2006) 385–398
[17] Automatic composition of semantic web services. Zhang, R., Arpinar, I.B., Aleman-Meza, B.: In: 1st International Conference on Web Services. (2003) 38–41, L´ecu´e, F., L´eger, A.: A formal model for semantic web service composition. In: ISWC 2006. LNCS, vol. 4273 (2006) 385–398
[18] Semi-automatic composition of web services using semantic descriptions. Sirin, E., Hendler, J.A., Parsia, B., In: 1st Workshop on Web Services: Modeling, Architecture and Infrastructure. (2003) 17–24.
[19] Wnt pathway curation using automated natural language processing: combining statistical methods with partial and full parse for knowledge extraction. Carlos Santos, Daniela Eggle, and David J. States. Bioinformatics, 21(82005):1653 – 1658, November2005.
[20] Service-Oriented Computing: Semantics,Processes, Agents. Munindar P. Singh and Michael N. Huhns, John Wiley & Sons, 2005
[21] A petri net-based model for web service composition. In CRPITS'17: Proceedings of the Fourteenth Australasian database conference on Database technologies 2003, pages 191–200, Rachid Hamadi and Boualem Benatallah. Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc
[22] Design and analysis of workflow processes with petri nets. C. Coves, D. Crestani, and F. Prunet. IEEE International Conference on Systems, Man, and Cybernetics,1:101–106, 1998, W. Derks, J. Dehnert, P. Grefen, and W. Jonker. Customized atomicity specification for transactional workflows. The Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications, CODAS2001, pages 140–147, 2001.
[23] Time petri nets for workflow modelling and analysis. Sea Ling and Heinz Schmidt. IEEE International Conference on Systems, Man, and Cybernetics, 4:3039–3044, 2000.
[24] Generic workflow models: how to handle dynamic change and capture management information? Proceedings of 1999 IFCIS International Conference on Cooperative Information Systems, W.M.P van der Aalst. pages 115–126, 1999.
[25] R. Milner. The polyadic _-calculus: A tutorial. Logic and Algebra of Specification, pages 203 – 246, 1993.
[26] Web Services Flow Language (WSFL). Technical report, IBM. http://xml.coverpages.org/wsfl.html.,Frank Leymann. Web Services Flow Language (WSFL 1.0). Technical report, IBM,May 2001.
[27] Change sequence mining for interdependent context aware service processes using partial derivatives. B. Jacob and K. Promod. In Nature Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, pages 1486 –1491, dec. 2009
[28] A Survey of Web Service Mining Techniques Gourav Mitra and Varun Goyal
[29] Research challenges and opportunities in web services mining. W. Gaaloul, S. Bhiri, and C. Godart.
[30] "An Approach for QoS-aware Service Composition based on Genetic Algorithms, "Proceeding GECCO '05 Proceedings G. Canfora, M. Di Penta, R. Esposito, M. Luisa Villani
[31] "A survey on web services composition," S. Dustdar, W. Schreiner.Journal International Journal of Web and Grid Services , Vol. 1, Issue 1, August 2005.
[32] "Flow- Based Service Selection for Web Service Composition Supporting Multiple QoS Classes," V. Cardellini, E. Casalicchio, V. Grassi,; F. Lo Presti Web Services ICWS 2007, IEEE , pp. 743 – 750, July 2007.
[33] Semantic Markup for Web Services. Martin, D., et al., OWL-S. 2004: http://www.w3.org/Submission/OWL-S/.
[34] UDDI Spec Technical Committee, UDDI Version 3.0.2. 2004: http://www.uddi.org.
[35] A Goal Specification Language for Automated Discovery and Composition of Web Services, Agarwal, S., in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 07). 2007, IEEE Computer Society: Silicon Valley, USA. p. 528—534.
[36] Web services: concepts, architectures and applications. Alonso, G., Casati, F., Kuno, H., Machiraju, V. Springer-Verlag (2004)
[37] Semantically enabled service-oriented architectures: a manifesto and a paradigm shift in computer science. Brodie, M.L., Bussler, C., de Bruijn, J., Fahringer, T., Fensel, D., Hepp, M., Lausen, H., Roman, D., Strang, T., Werthner, H., Zaremba, M.: Technical report, DERI (2005)
[38] Cardoso and Sheth 2002; Chandrasekaran et al. 2002.
[39] 'X Methods', Hong, T. and Hong, J. (2000) , http://www.xmethods.net, accessed 22 September 2002.
[40] Performance of SOAP in Web Service Environment Compared to CORBA, Elfwing, R., Paulsson, U., and Lundberg, L. In Proceedings of the Ninth Asia-Pacific Software Engineering Conference, IEEE, 2002.
[41] Performance of SOAP Implementations, Davis, D., and Parashar, M., Latency. In Proceedings of 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE, 2002.
[42] Performance Comparison of CORBA and RMI, Information and Software Technology 42, Juric, M.B., Rozman, I., and Hericko. M. pp 915-933, 2000.
[43] Java 2 Distributed Object Models Performance, Analysis, Comparison, and Optimization, Juric, M.B, Rozman, I., Stevens, A.P., Hericko, M.and Nash, S., In Proceedings of 7th International Conference on Parallel and Distributed System, IEEE, 2000.